

CLAIMS

What is claimed is:

- 1 1 A system comprising:
 2 a bus;
 3 the bus is connected to a first device and a second device;
 4 a data structure within a first device which has a hierarchy of descriptors
 5 containing at least a first list descriptor and a second list descriptor and at least one
 6 entry descriptor, each descriptor containing at least one data structure, and
 7 the second device using at least one data structure in a command.
- 1 2. The system of claim 1, wherein one of the first list descriptor and the second
 2 list descriptor has information about a first list.
- 1 3. The system of claim 1, wherein an unambiguous specification of a data
 2 structure is made.
- 1 4. The system of claim 2, wherein the information about the first list is placed in
 2 a second list.
- 1 5. The system of claim 4, wherein the second list has a beginning and an end;
 2 and
 3 the information is placed at the end of the second list.
- 1 6. The system of claim 4, wherein the information from the first list is placed in
 2 the second list in an extended_information field.
- 1 7. The system of claim 6, wherein the second device accesses the
 2 extended_information field allowing the controller to move backwards in a
 3 hierarchy.
- 1 8. A method comprising:
 2 coupling a first device and a second device to a bus;

3 placing into a data structure a descriptor specifier which specifies an entry by
4 a list identifier and an object identifier.

1 9. The method of claim 8, further comprising:
2 opening a data structure by the first device; and
3 reading at least one entry in the data structure by a second device.

1 10. The method of claim 8, comprising:
2 embedding information about a parent entry within a child list descriptor.

1 11. The method of embedding information about a root_list_ID within a root list
2 descriptor, comprising:
3 reading the extended information field by a controller.

1 12. The method of claim 11, further comprising:
2 accessing the extended_information field allowing the second device to move
3 backwards in a data structure hierarchy.

1 13. The method of claim 8, further comprising:
2 placing the descriptor specifier in an extended_information field.

1 14. The method of claim 13, further comprising:
2 reading the information from the extended_information field.

1 15. The method comprising:
2 using a descriptor specifier that specifies an entry by list_ID and object_ID.

1 16. The method of claim 15, further comprising:
2 opening the descriptor using the descriptor specifier.

1 17. The method of claim 15, further comprising:
2 embedding information about a parent entry within the descriptor specifier.

1 18. The method of claim 15, further comprising:
2 placing the information about the parent entry at an end of a child list.

- 1 19. The method of claim 17, further comprising:
2 reading the information from an extended_information field.
- 1 20. The method of claim 19, further comprising:
2 using a descriptor specifier for opening a corresponding a parent entry.
- 1 21. A method comprising:
2 using a descriptor having a field of a list descriptor ID and an object ID.
- 1 22. The method of claim 21, further comprising:
2 placing the parent descriptor info block in an extended_information field.
- 1 23. The method of claim 21, further comprising:
2 accessing the extended_information field allowing a controller to move
3 backwards in the descriptor hierarchy.
- 1 24. The method of claim 21, further comprising:
2 embedding information about a parent entry within a list descriptor.
- 1 25. The method of claim 21, further comprising:
2 placing the information about the entry in a child list.
- 1 26. A method comprising:
2 embedding a parent descriptor info block within a list descriptor.
- 1 27. The method of claim 26, further comprising:
2 placing a descriptor specifier info block for the parent descriptor in one of a
3 root list descriptor and a child list descriptor.
- 1 28. The method of claim 27, wherein the root list has a first position and a second
2 position; and
3 the descriptor specifier is placed at the second position.
- 1 29. The method of claim 27, wherein the child list has a third position and a
2 fourth position; and

the descriptor specifier is placed at the fourth position.

30. The method of claim 27, further comprising:
using a descriptor specifier in a descriptor command for opening a parent entry.

31. The method of claim 30, further comprising:
navigating descriptors in a descriptor hierarchy wherein navigating is in a backward direction.

32. A method comprising:
using a delete descriptor command, the delete descriptor command is configured to delete one of a root list, a child list, and an entry.

33. The method of claim 32, further comprising:
deleting a child_ID in a descriptor hierarchy.

34. The method of claim 33, further comprising:
updating has_child_ID attributes.

35. The method of claim 34, further comprising:
updating entry_descriptor_length.

36. The method of claim 35, further comprising:
updating list_descriptor_length.

37. The method of claim 32, further comprising:
deleting a child list descriptor in a descriptor hierarchy.

38. The method of claim 37, further comprising:
deleting a parent entry descriptor.

39. The method of claim 38, further comprising:
updating no_of__entry descriptors.

1 40. The method of claim 39, further comprising:
2 updating a list_descriptor_length.

1 41. The method of claim 40, further comprising:
2 deleting a first child list descriptor in a descriptor hierarchy.

1 42. The method of claim 41, further comprising:
2 deleting a second child list descriptor.

1 43. The method of claim 42, further comprising:
2 deleting a list descriptor.

1 44. A system comprising:
2 a bus;
3 the bus is connected to a first device and a second device;
4 a data structure within a first device which has a hierarchy of descriptors
5 containing a child list descriptor which is deleted by the second device, the second
6 device uses a delete descriptor, the delete descriptor command is configured to
7 delete one of a root list, a child list, and an entry.;
8 the second device deletes a child_ID; and
9 one of a has_child_ID attribute, an entry_descriptor_length, and the
10 list_descriptor_length is updated.